Attorney Docket No.: 40152-000400US

PROVISIONAL PATENT APPLICATION

AUTHENTICATION OF CONTENT DOWNLOAD

Inventor(s): Nathan F. Raciborski, a citizen of the United States, residing at,
2643 East Spring Rd.
Phoenix, AZ 85032

Assignee: Limelight Networks, LLC
8936 North Central Avenue
Phoenix, AZ 85020

Entity: Small entity

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 303-571-4000

AUTHENTICATION OF CONTENT DOWNLOAD

[01] This application incorporates by reference in its entirety US Provisional Patent Application Serial No. 60/___,___ filed on July 28, 2003, entitled MULTIPLE OBJECT DOWNLOAD, referenced by Attorney Docket No. 40152-000200US.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[02] The ensuing description provides preferred exemplary embodiment(s) only, and is not intended to limit the scope, applicability or configuration of the invention. Rather, the ensuing description of the preferred exemplary embodiment(s) will provide those skilled in the art with an enabling description for implementing a preferred exemplary embodiment of the invention. It being understood that various changes may be made in the function and arrangement of elements without departing from the spirit and scope of the invention as set forth in the appended claims.

[03] Today, files are typically downloaded using Hyper Text Transfer Protocol (HTTP). The origin server initiates the download after a file is requested. The file is sequentially sent in packets to the client computer upon request. Where packets do not arrive at the client computer, they are requested again from the origin server. The origin server presumes the downloaded file has been successfully sent after the client computer stops requesting packets. The client computer stops requesting packets after enough packets are received to equal the file size.

[04] In many cases, the file is not actually stored to the client computer in a usable form. For example, the client computer could crash before it could request a corrupt packet, a virus could attach to the file, an error could occur when writing the file to the hard drive and/or other problems that prevent the client computer from using the file. When users are unable to use a downloaded file, they may contact the operator of the origin server to get authorized to download another copy of the file.

[05] In one embodiment, the client computer installs a download manager program. In other embodiments, the functionality of the download manager program could be integrated into the operating system; web browser, content player, or other application software; or a browser script, applet or plug-in. The download manager checks the stored version of the file to authenticate it. Upon successful authentication, the origin server is notified such that delivery is assured.

[06] Authentication is a process where the veracity of the file is confirmed. In one embodiment, metadata in the file indicates a CRC, hash or checksum of an authentic file. The metadata could use XML or other format. For example, the file could include a Secure Hashing Algorithm 1 (SHA-1) hash in XML that is checked against a SHA-1 hash calculated by the download manager. Some embodiments may query the origin server for the hash value of a particular file where the hash value is not stored in metadata. A database at the client computer, origin server or other location could store the hash values for files in a manner such that the download manager can access the hash values when checking authenticity.

[07] Where the file cannot be authenticated, a replacement file could be retrieved without contacting customer support for the origin server. The download manager notifies the user that the download was unsuccessful. The user is given the option to download the file immediately or to request the download later. In some cases, the user's account may not be charged or any charge may be reversed where the download is unsuccessful. Other embodiments, may automatically download a replacement file until an authentic file is confirmed as received. If a user does call customer support complaining of a corrupt file, a database can be queried that the download was successful and that the user once had a valid copy of the file.

[08] Some embodiments of the invention could periodically check all downloaded files to authenticate them. Where one or more files cannot be authenticated, the files could be replaced. File updates can be distributed by updating a database of hash values with a hash value of a new version of the file. When the hash value cannot be authenticated at the next check, the download manager will download the new version and overwrite the old version. For example, the origin server operator may realize that a particular music file has an encoding error and replace it with a corrected version. All users that downloaded the music file with the error can receive the new one when the download manager performs the next authentication.

[09] In some cases, copyright holders do not want the users to create derivative works or otherwise modify a file. After modification, the download manager notices that the hashes no longer match and will update the file with the original version. Some embodiments may notify the user before any downloaded file is overwritten. Other embodiments may just delete the modified file where the hash no longer matches. Examples of downloaded files include digitized video, digitized audio, digitized sound, digital pictures, software, electronic books, electronic documents, or other electronic files.

[10] Other programs could authenticate files. For example, a virus checking program could verify the authenticity of each file. Where files are found to be corrupt because of a virus or other problem, the origin server could be contacted for another copy. The database or XML with the hash value could also store information for contacting the origin server for another copy of the file. Some embodiments could check authorization to determine if the client computer is authorized to get replacements for the file. Some embodiments, may charge for the replacement or provide it for free. For example, where the software file is covered under a service contract or content subscription, replacements are allowed, but are charged for where the contract or subscription has expired.

[11] While the principles of the invention have been described above in connection with specific apparatuses and methods, it is to be clearly understood that this description is made only by way of example and not as limitation on the scope of the invention.